Report dall'ultimo Lustre User Group: esperienze di grandi centri di calcolo e prospettive

Giacinto DONVITO
Università e INFN -- BARI

Outlook

- *Lustre:*
  - *History*
  - *Status*
  - *Future*
- *Reports from sites@LUG2011*
- *Interesting hints@LUG2011*
- *Emergency plan*
- *Conclusions*

Lustre: History

- *Lustre 1.0 released in 2003 by Cluster File Systems (founded by Peter Braam, n.d.r)*
- *Cluster File Systems was acquired by Sun Microsystems in October 2007*
- *November 2008, Braam left Sun Microsystems*
- *April 2009: SUN was acquired by Oracle Corporation*
- *Lustre 2.0.0, released in August 2010*
- *September 2010: Oracle will not continue developing 2.x lustre tree*
- *End 2010 beginning 2011 few Open source community born in order to go on developing Lustre 2.1*

About Lustre

November 2010: Xyratex hired Peter Braam

"Lustre is recognized as a leading high performance clustered file system in High Performance Computing with over 60% share of the Top 100 systems in the Top 500," said Earl Joseph, Research Analyst at IDC. "Peter Braam and Peter Bojanic are recognized as key leaders of the Lustre community and by reuniting them, there's no question that this is a very positive move for the broader HPC community and that it will help to ensure that Lustre will continue to be a key element of HPC data storage environments."

Open source communities

- *HPCFS (www.hpcfs.org):*
  - *Members: FERMILAB, XIOTECH, WHAM CLOUD, INTEL, PSEC, SGI, NASA, PNL, CHEVRON/TEXACO, SLAC-STANFORD, INDIANA UNIVERSITY, SANDIA, LBL, NRL, MELLANOX, ROUTING DYNAMICS, HP, DELL*
  - *status: will merge with opensfs*
- *OpenSFS (www.opensfs.org):*
  - *Members: LLNL, ORNL, DDN, Cray, SGI*
- *EOFS (www.eofs.org):*
  - *European Initiative*
  - *Members: Bull, CEA, Data Direct Networks, Forschungszentrum Jülich, GSI, Hewlett-Packard, HPCFS, Leibniz Rechenzentrum, ParTec CCC, T-Platforms, Universität Zürich, Universität Paderborn, Whamcloud, EUROTECH, Xyratex Technology Limited*

Release status

- *2.1 Release coming:*
  - *Whamcloud had much of the necessary infrastructure in place to create Lustre releases and so volunteered to host 2.1 community release (git, JIRA, gerrit, jenkins, maloo etc)*
  - *2.1 Release meetings open to anyone*
  - *Whamcloud contributor agreement means that no single organization will ever hold Lustre copyright again*
  - *All three community groups (EOFS, HPCFS and OpenSFS) are in support of this approach*

Release status

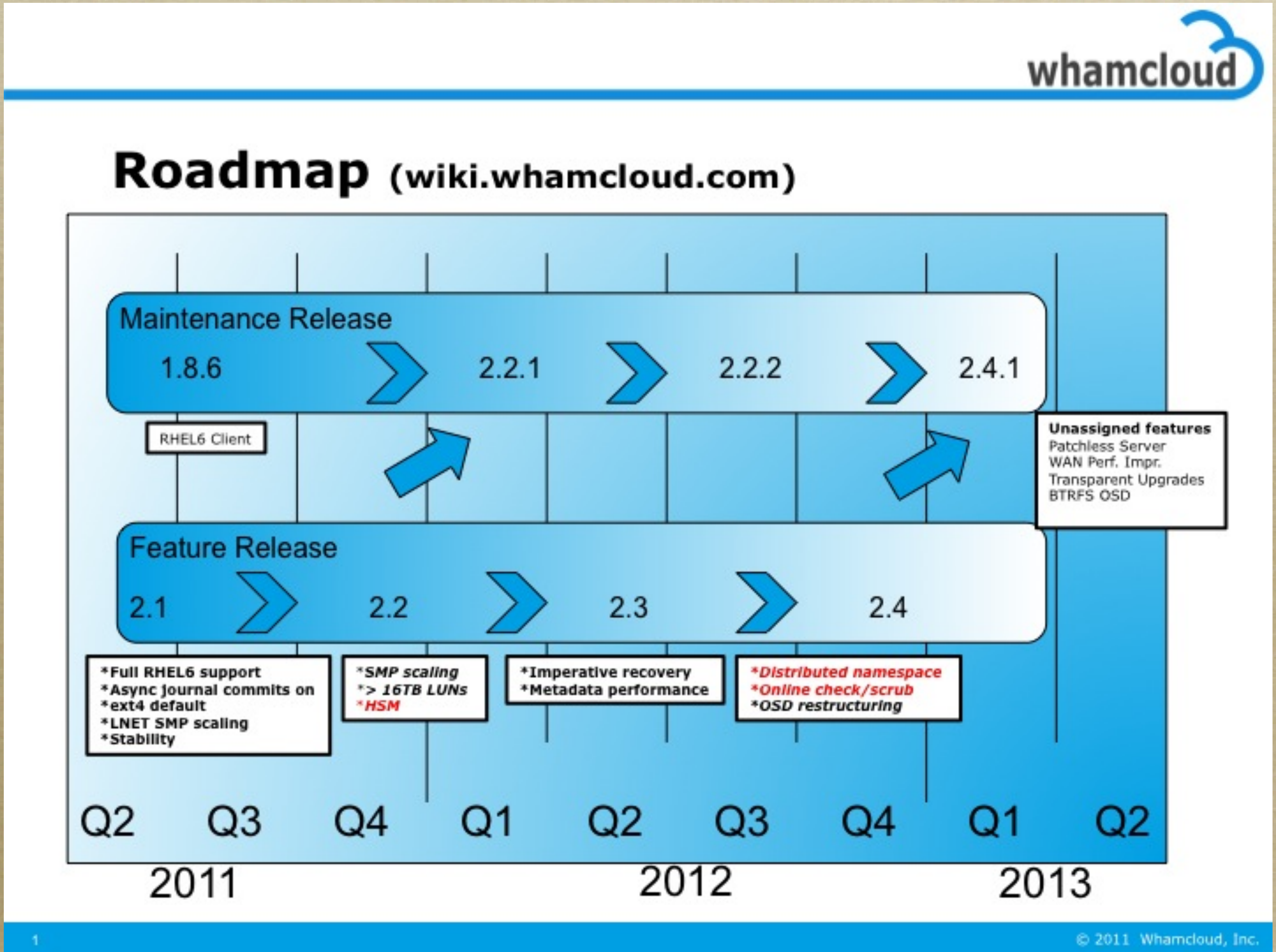
- *2.1 Release coming:*
  - *RHEL6 Server and Client support*
  - *Async journal commits by default*
    - *Added in 1.8.2; turned off by default*
    - *Already used in production at many sites (LLNL, ORNL, DDN sites)*
  - *Ext4 by default*
  - *2.x performance to match\exceed 1.8.x*
    - *SMP Scaling\IO performance*

Release status

- *Interest and collaboration*
    - *Broad community interest in release*
        - *25 different organizations registered on mailing list (at least)*
        - *11 different organizations represented at meetings*
        - *3 different organizations submitted patches (LLNL, ORNL, Xyratex)*
        - *4 different organizations offered to help with testing (Bull, Cray, LLNL, ORNL)*
- *Lustre 2.1 is relatively simple*
    - *The scope of the release was already defined and most of the work was done*
- *It will be much harder to manage release content for future releases across multiple stakeholder groups*
    - *Need to find a workable long-term model*
- *Whamcloud will be producing future 2.x releases for its customers*
    - *Core Lustre code will be open to all and available to any other releases*

Lustre: Future of the releases

- *Oracle will continue to produce Lustre 1.8.x releases*
  - *Lustre 2.1 due out this summer*
  - *Lustre 2.x releases TBD*



whamcloud

Roadmap (wiki.whamcloud.com)

Maintenance Release: 1.8.6 → 2.2.1 → 2.2.2 → 2.4.1

RHEL6 Client

Unassigned features
Patchless Server
WAN Perf. Impr.
Transparent Upgrades
BTRFS OSD

Feature Release: 2.1 → 2.2 → 2.3 → 2.4

*Full RHEL6 support
*Async journal commits on
*ext4 default
*LNET SMP scaling
*Stability

*SMP scaling
*> 16TB LUNs
*HSM

*Imperative recovery
*Metadata performance

*Distributed namespace
*Online check/scrub
*OSD restructuring

© 2011 Whamcloud, Inc.

Report from sites:
National Climate Computing Center

- *Capacity: fit the use cases that need performance*
    - *Scratch*
    - *Hot dataset cache*
    - *Semi-persistent library*
    - *Staging and buffering for WAN transfer*
- *Consistency: use cases increase variability*
    - *Some demand capability (scratch, hot cache)*
        - *Significantly more random access*
    - *Some are more about capacity (library, staging)*
        - *More sequential access*
- *Cost: Always an issue*
    - *On a fixed budget, I/O robs compute*
    - *Capability costs compute resources (more I/O nodes)*

- *Phase 1: Cray XT6*
    - *2,576 AMD Opteron 6174*
- *Phase 2: Cray XE6*
    - *5,200 AMD Opteron 16-core*

Report from sites:
National Climate Computing Center

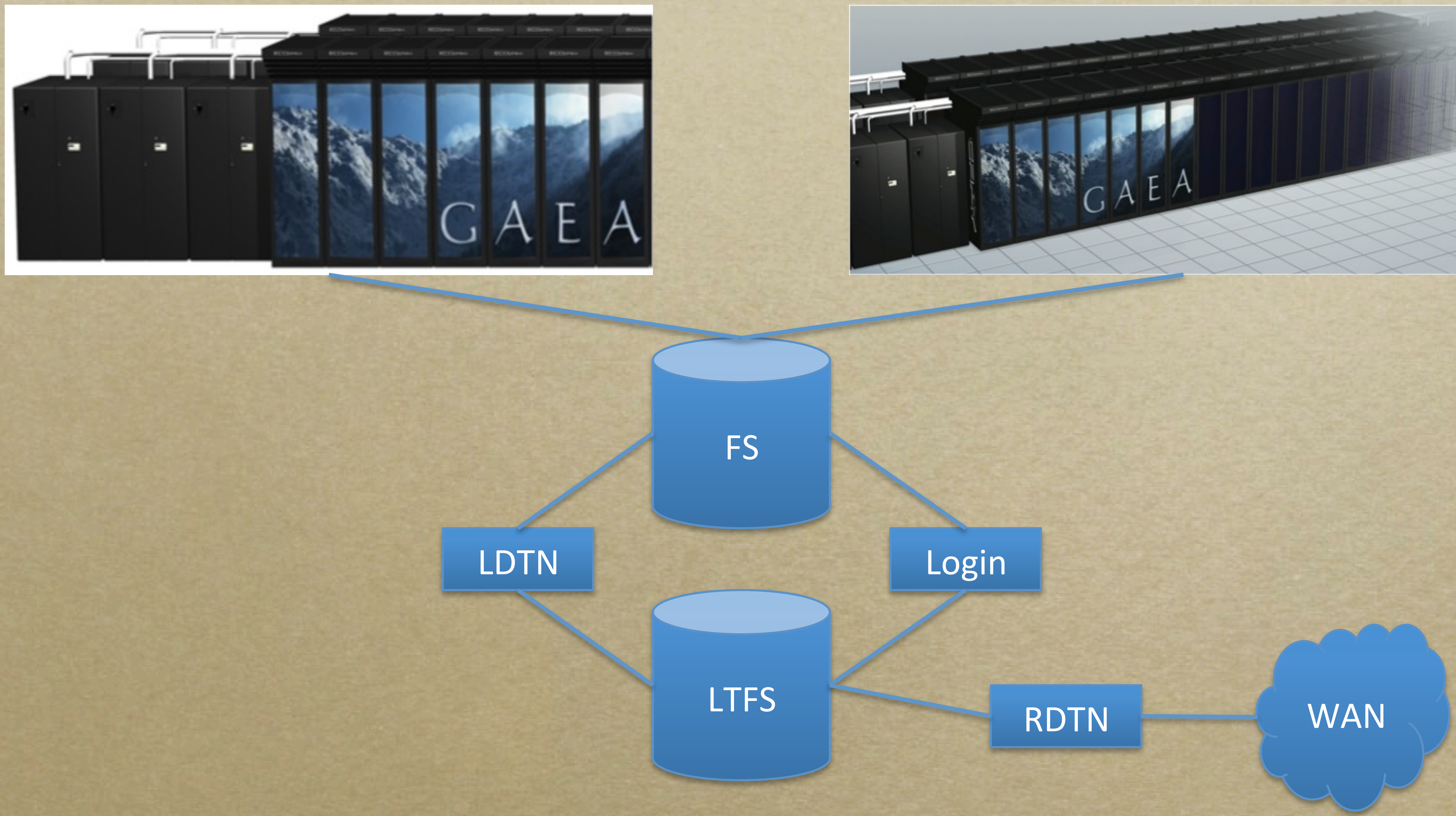
- *Fast Scratch*
  - *18x DDN SFA10000*
  - *2,160 active 600GB SAS 15000 RPM disks*
  - *36 OSS*
  - *InfiniBand QDR*

120 Disk per DDN system

- *Long Term Fast Scratch*
  - *8x DDN SFA10000*
  - *2,240 active 2TB SATA 7200 RPM disks*
  - *16 OSS*
  - *InfiniBand QDR*

280 Disk per DDN system

Report from sites:
National Climate Computing Center



Gaea filesystem architecture

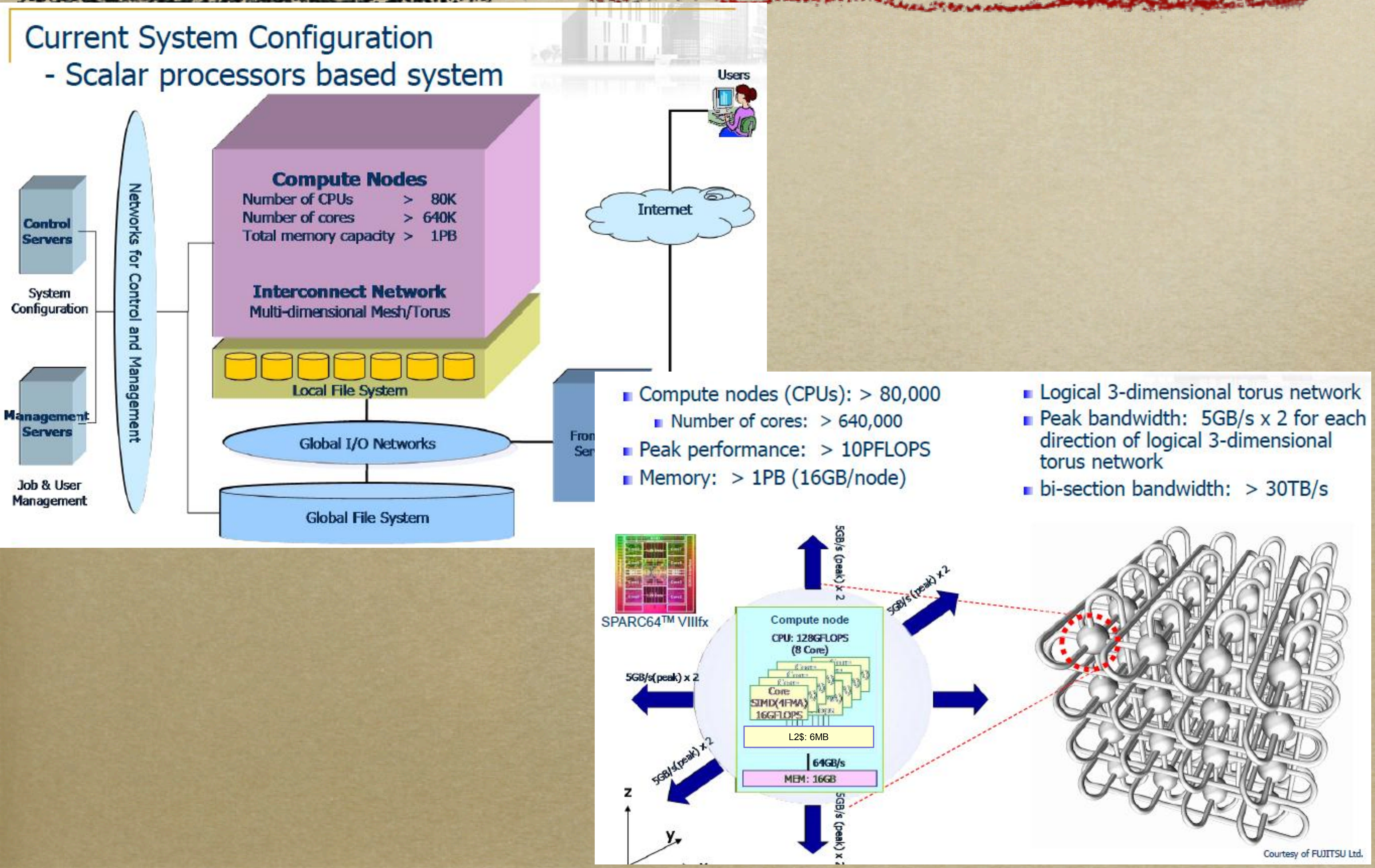
Report from sites:
National Climate Computing Center

- *User prospective:*
  - *Performance*
    - *Model initialization took 15 mins before now it takes 8 mins.*
  - *Reliability*
    - *Generally it's a reliable and stable filesystem.*
  - *Size*
    - *Scalability allow for large filesystems, less data movement, and larger experiments.*

Report from sites:
National Climate Computing Center

- *User prospective:*
  - *They are not insulated from the bad practices of misbehaving users.*
  - *Do not have the necessary tools to manage the filesystems and user behavior.*
    - *Quotas*
    - *Slowness and potential issues with using standard unix commands*
      - *du, ls, find, etc.*
  - *Confused with problems in their jobs resulting from OST or OSS failures.*
  - *Users don't know if the I/O error they receive in their output is permanent or transient.*
  - *If parts of the filesystem are offline, users and management want the ability to quickly see this and adjust the running workload to it.*
    - *Ideally, this would be automated.*

An Overview of Fujitsu's Lustre Based File System



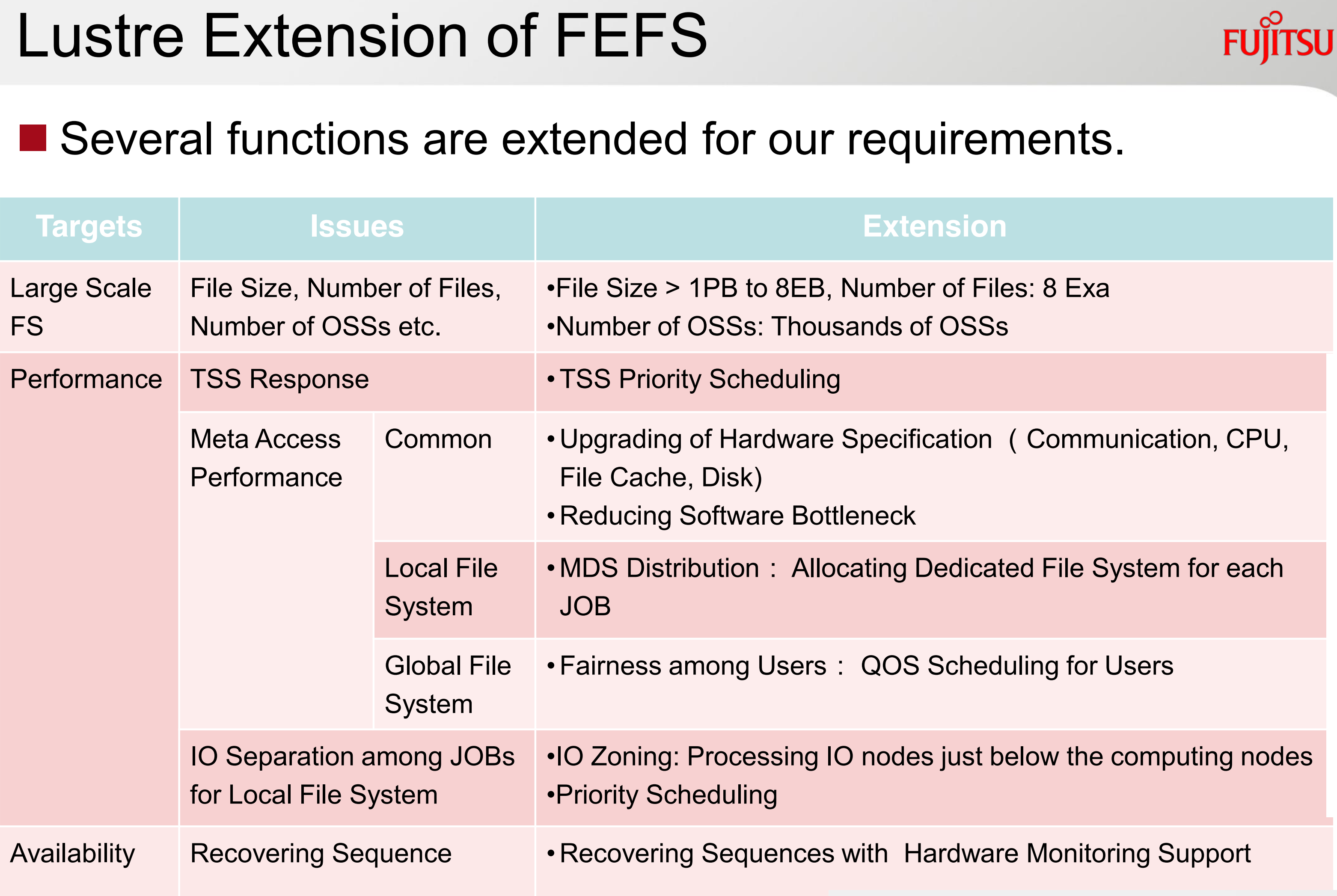
Current System Configuration
- Scalar processors based system

- Compute nodes (CPUs): > 80,000
  - Number of cores: > 640,000
- Peak performance: > 10PFLOPS
- Memory: > 1PB (16GB/node)

- Logical 3-dimensional torus network
- Peak bandwidth: 5GB/s x 2 for each direction of logical 3-dimensional torus network
- bi-section bandwidth: > 30TB/s

An Overview of Fujitsu‘s Lustre Based File System

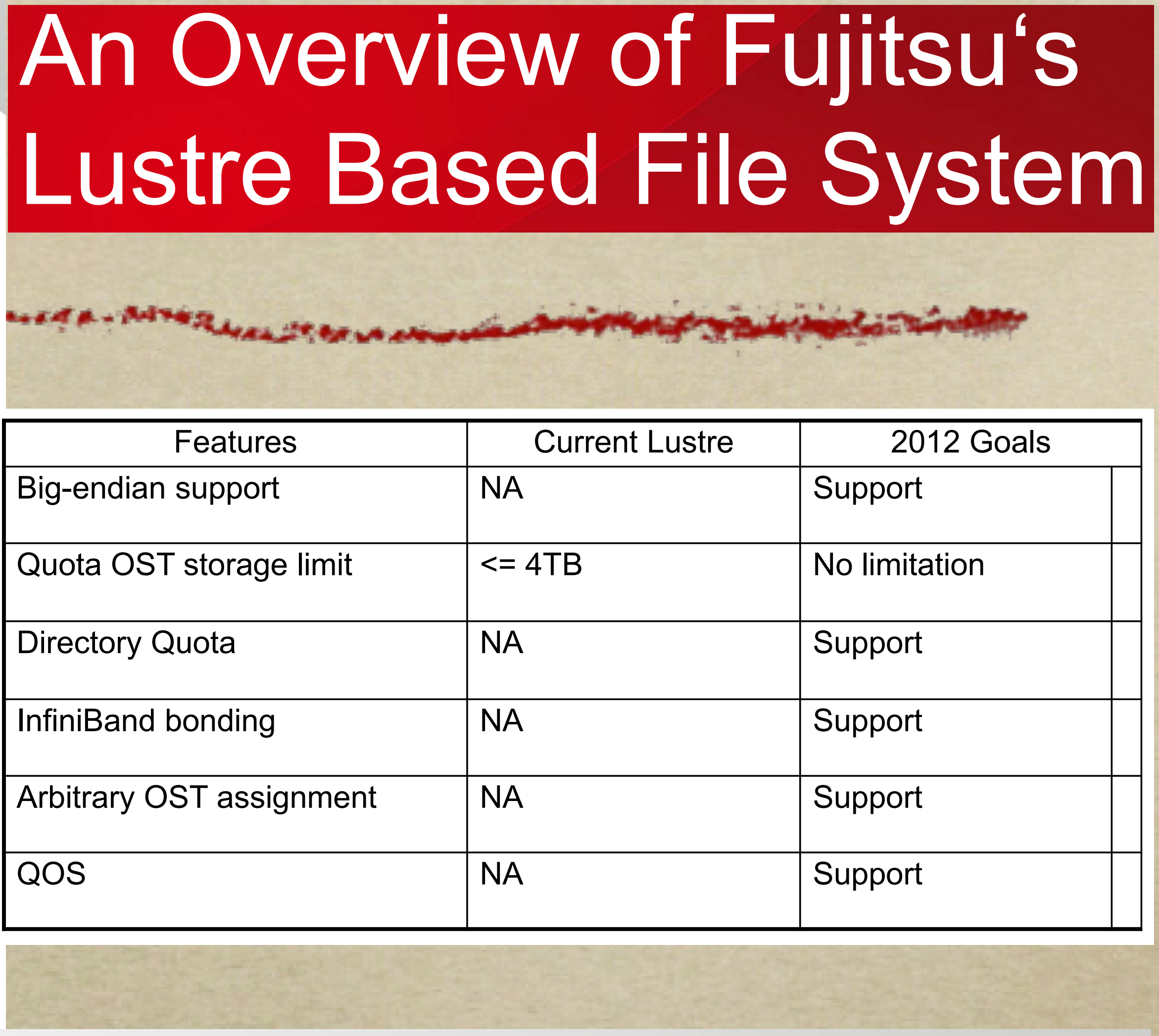
- *Goals of Fujitsu’s Cluster File System: FEFS*
- *FEFS(Fujitsu Exabyte File System) for peta scale and exa-scale supercomputer will achieve:*
- *Extremely Large*
  - *Extra-large volume (100PB~1EB).*
  - *Massive number of clients (100k~1M) & servers (1k~10k)*
- *High Performance*
  - *Throughput of Single-stream (~GB/s) & Parallel IO (~TB/s)*
  - *Reducing file open latency (~10k ops)*
  - *Avoidance of IO interferences among jobs.*
- *High Reliability and High Availability*
  - *Always continuing file service while any part of system are broken down.*
- *FEFS is optimized for utilizing maximum hardware performance by minimizing file IO overhead, and based on Lustre file system.*



Lustre Extension of FEFS

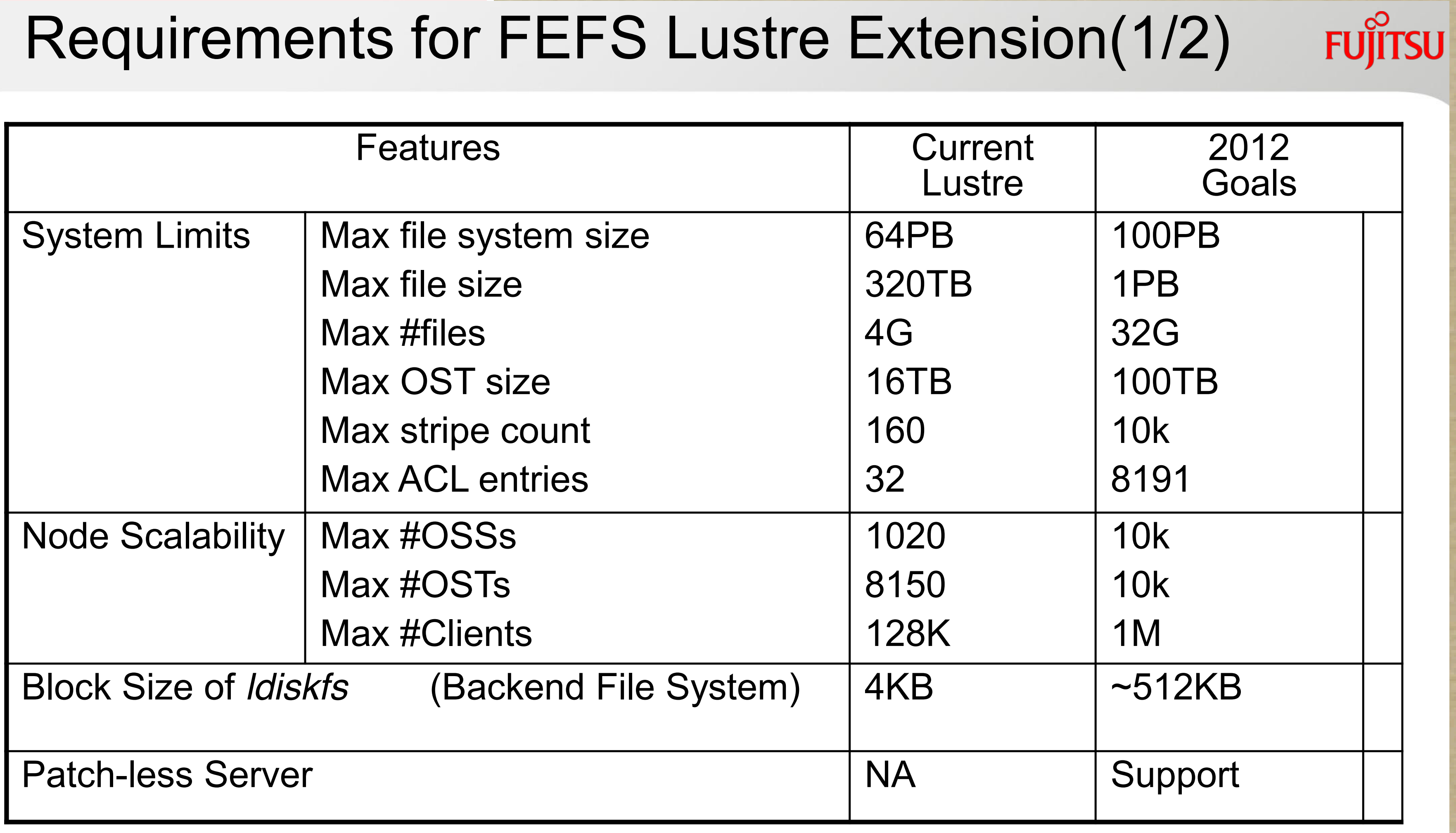
- Several functions are extended for our requirements.

Targets	Issues		Extension
Large Scale FS	File Size, Number of Files, Number of OSSs etc.		•File Size > 1PB to 8EB, Number of Files: 8 Exa •Number of OSSs: Thousands of OSSs
Performance	TSS Response		•TSS Priority Scheduling
	Meta Access Performance	Common	•Upgrading of Hardware Specification（Communication, CPU, File Cache, Disk) •Reducing Software Bottleneck
		Local File System	•MDS Distribution：Allocating Dedicated File System for each JOB
		Global File System	•Fairness among Users：QOS Scheduling for Users
	IO Separation among JOBs for Local File System		•IO Zoning: Processing IO nodes just below the computing nodes •Priority Scheduling
Availability	Recovering Sequence		•Recovering Sequences with Hardware Monitoring Support



An Overview of Fujitsu's Lustre Based File System

Features	Current Lustre	2012 Goals
Big-endian support	NA	Support
Quota OST storage limit	<= 4TB	No limitation
Directory Quota	NA	Support
InfiniBand bonding	NA	Support
Arbitrary OST assignment	NA	Support
QOS	NA	Support



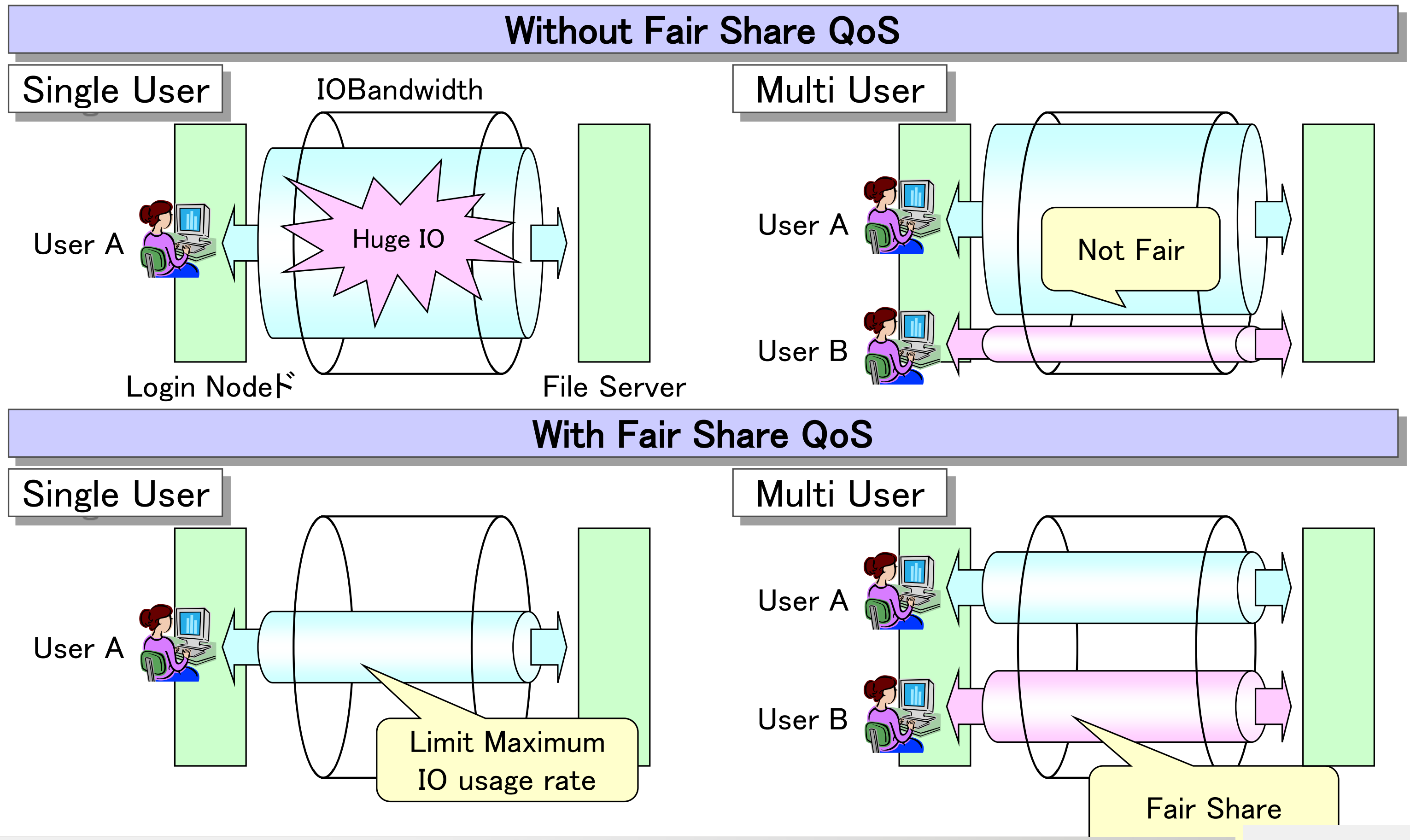
Requirements for FEFS Lustre Extension(1/2)

Features		Current Lustre	2012 Goals
System Limits	Max file system size	64PB	100PB
	Max file size	320TB	1PB
	Max #files	4G	32G
	Max OST size	16TB	100TB
	Max stripe count	160	10k
	Max ACL entries	32	8191
Node Scalability	Max #OSSs	1020	10k
	Max #OSTs	8150	10k
	Max #Clients	128K	1M
Block Size of *ldiskfs* (Backend File System)		4KB	~512KB
Patch-less Server		NA	Support

An Overview of Fujitsu‘s Lustre Based File System

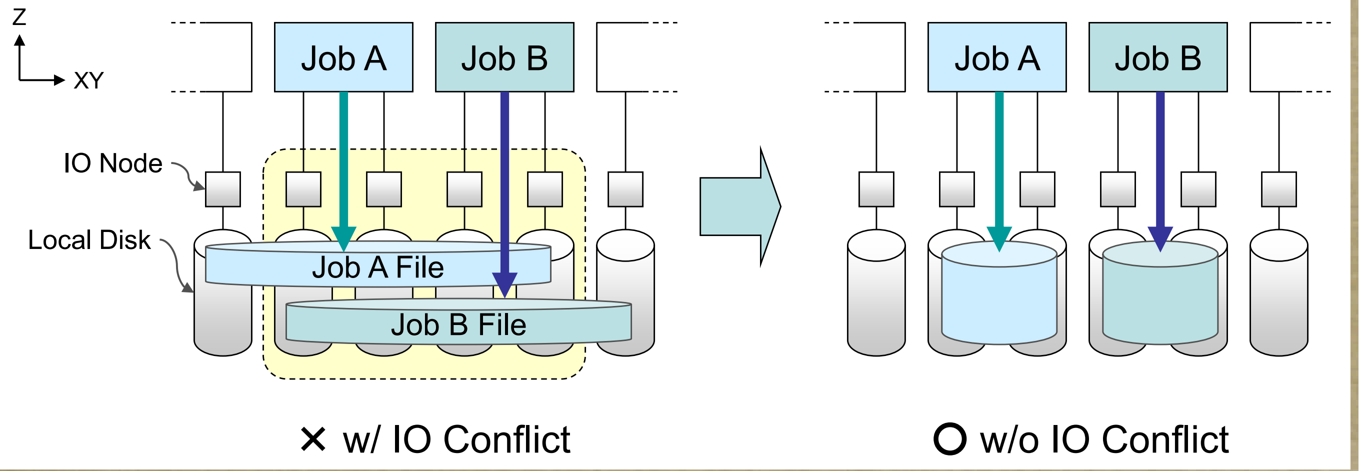
Fair Share QoS

- Avoiding from some one’s occupying file IO resources



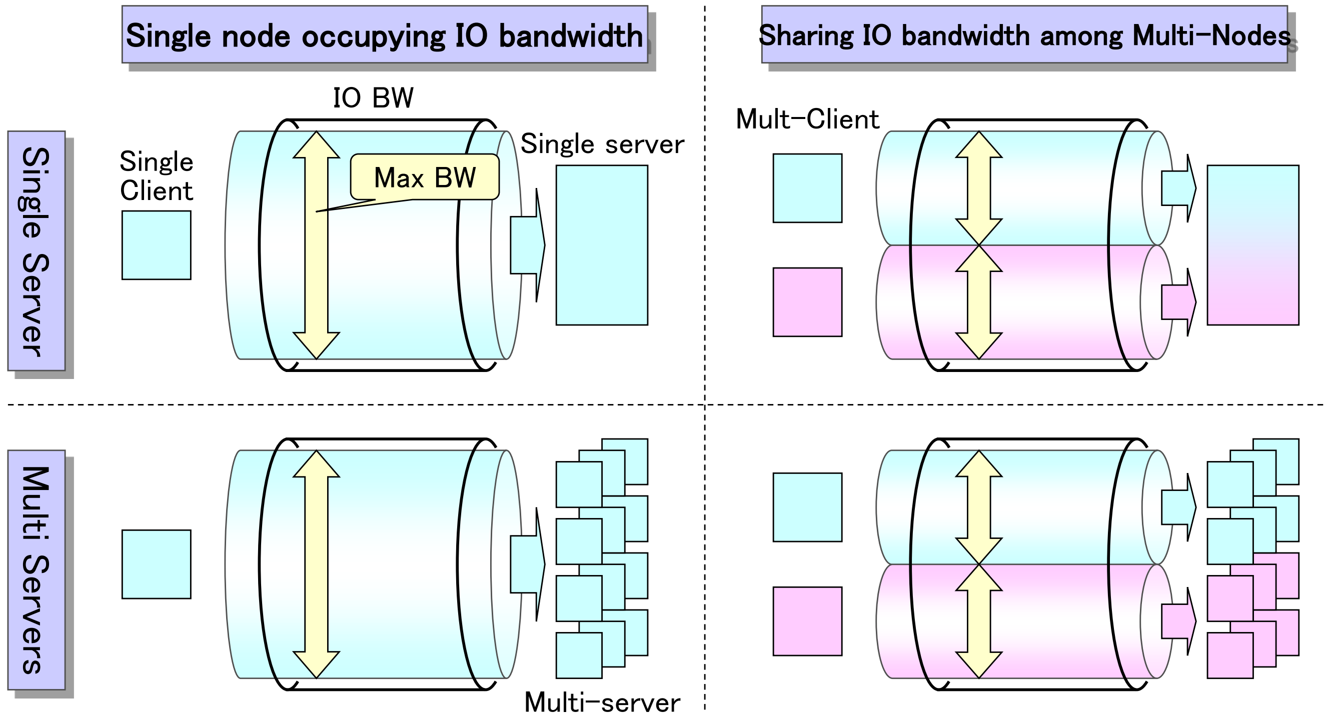
IO Zoning: IO Separation among JOBs

- Issue: Sharing disk volumes, network links among jobs cause IO performance degradation because of their conflicts.
- Our Approach: Separating of disk volumes, network links among jobs as much as possible.



Best Effort QoS

- Fair Share among users

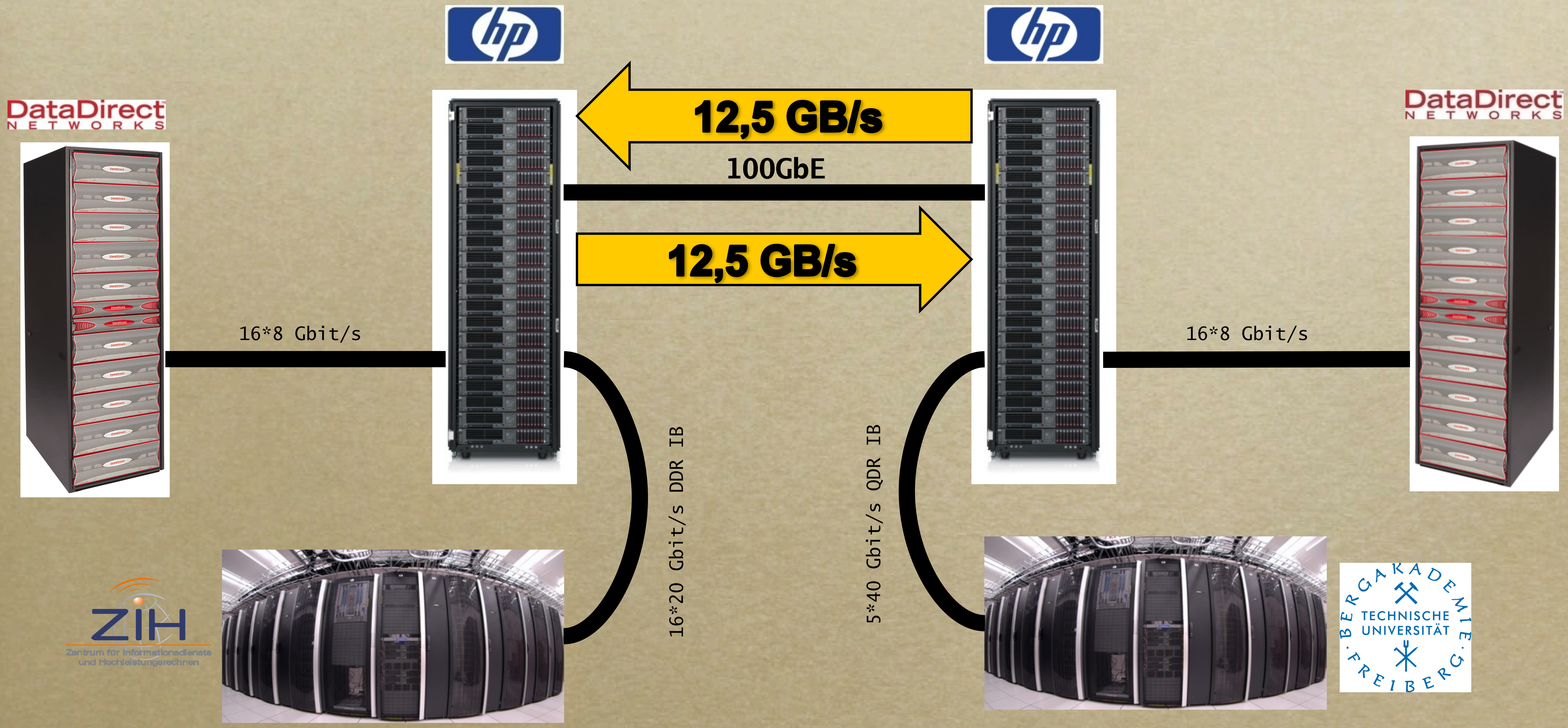


Lustre WAN @ 100GBit Testbed

Michael Kluge
michael.kluge@tu-dresden.de

Robert Henschel, Stephen Simms
{henschel,ssimms}@indiana.edu

Lustre WAN @ 100GBit Testbed



12,5 GB/s

100GbE

12,5 GB/s

16*8 Gbit/s

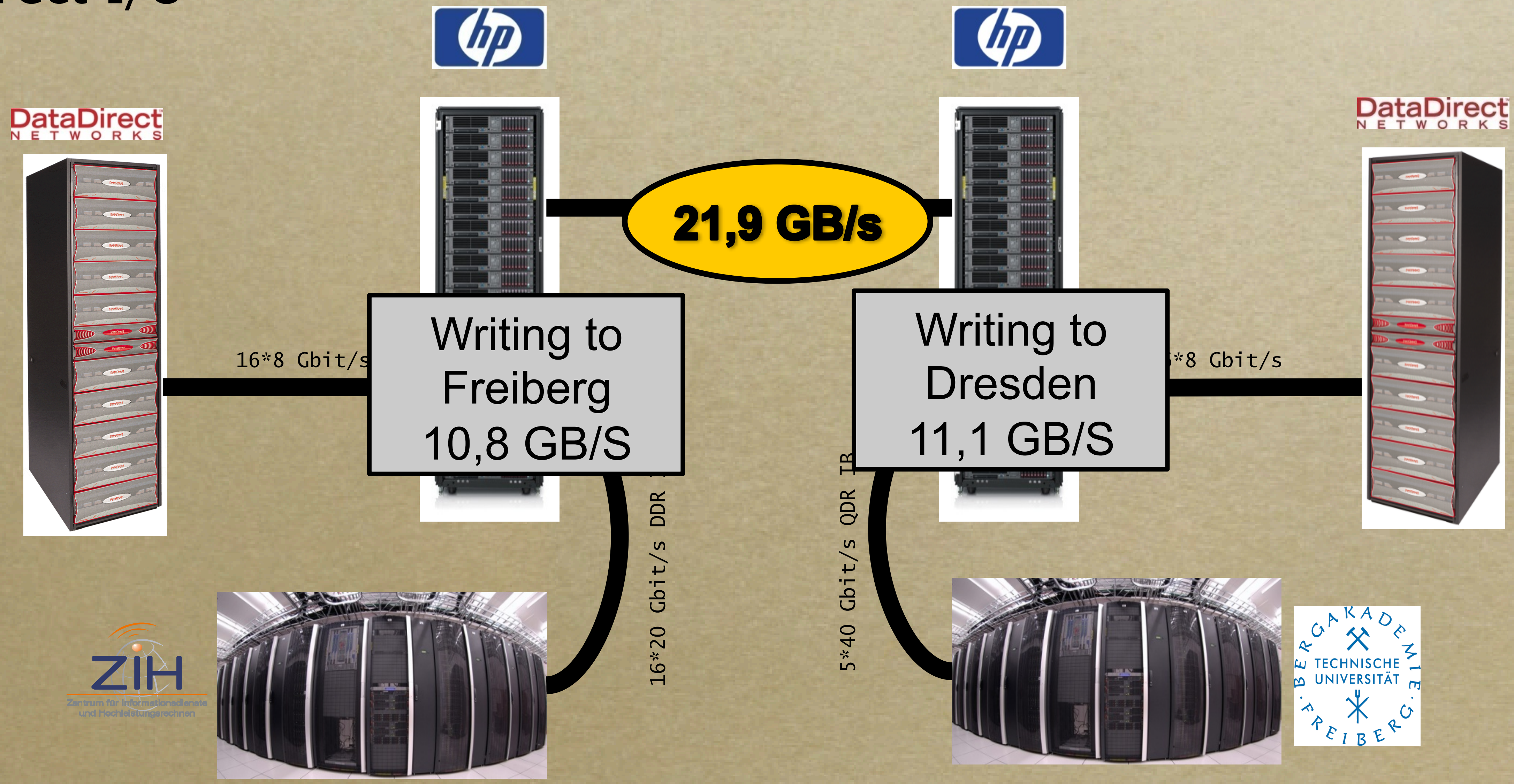
16*20 Gbit/s DDR IB

5*40 Gbit/s QDR IB

16*8 Gbit/s

Lustre WAN @ 100GBit Testbed

- 24 clients on each site
- 24 processes per client
- stripe size 1, 1 MiB block size
- Direct I/O



21,9 GB/s

Writing to Freiberg 10,8 GB/S

Writing to Dresden 11,1 GB/S

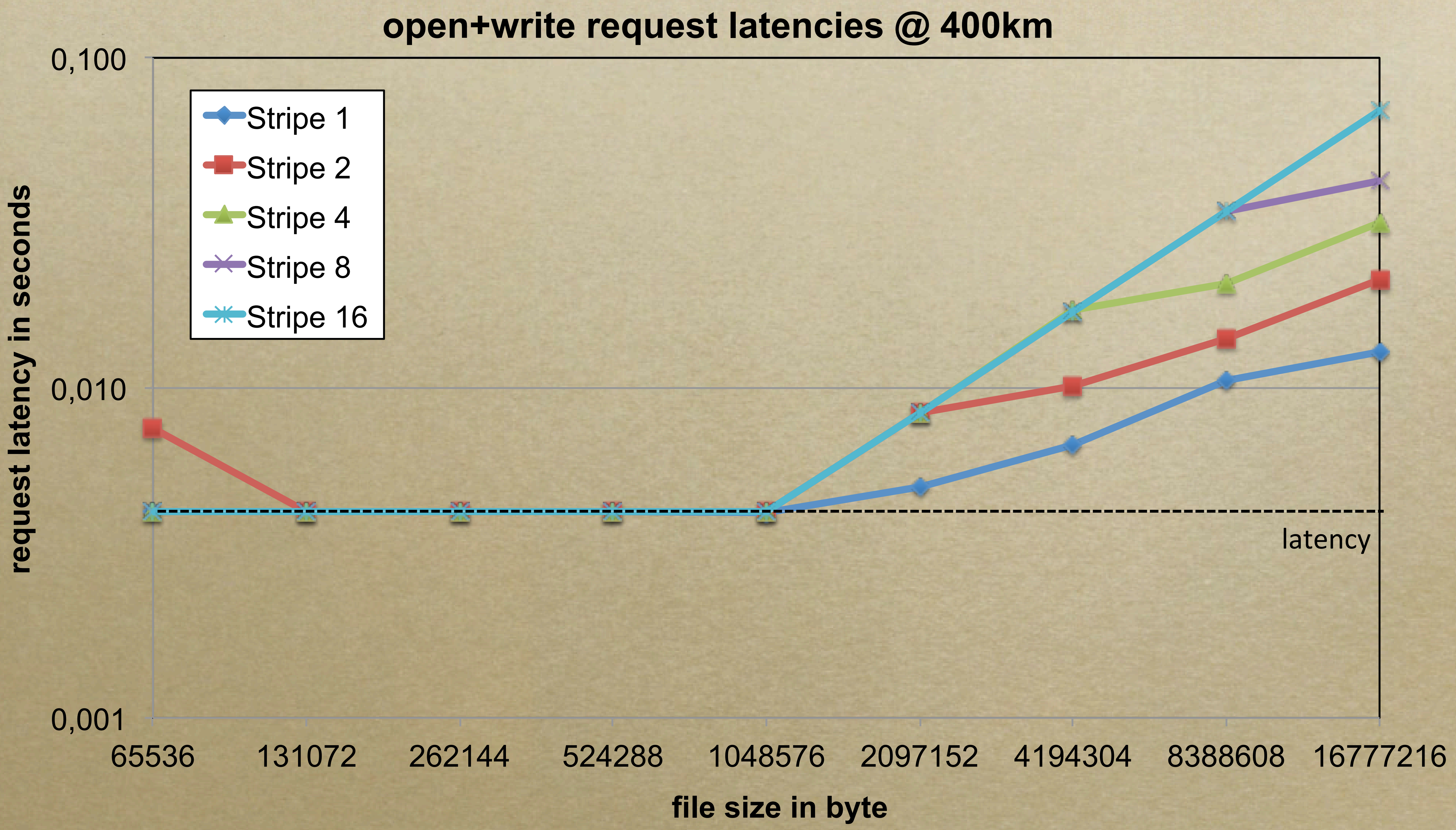
16*8 Gbit/s

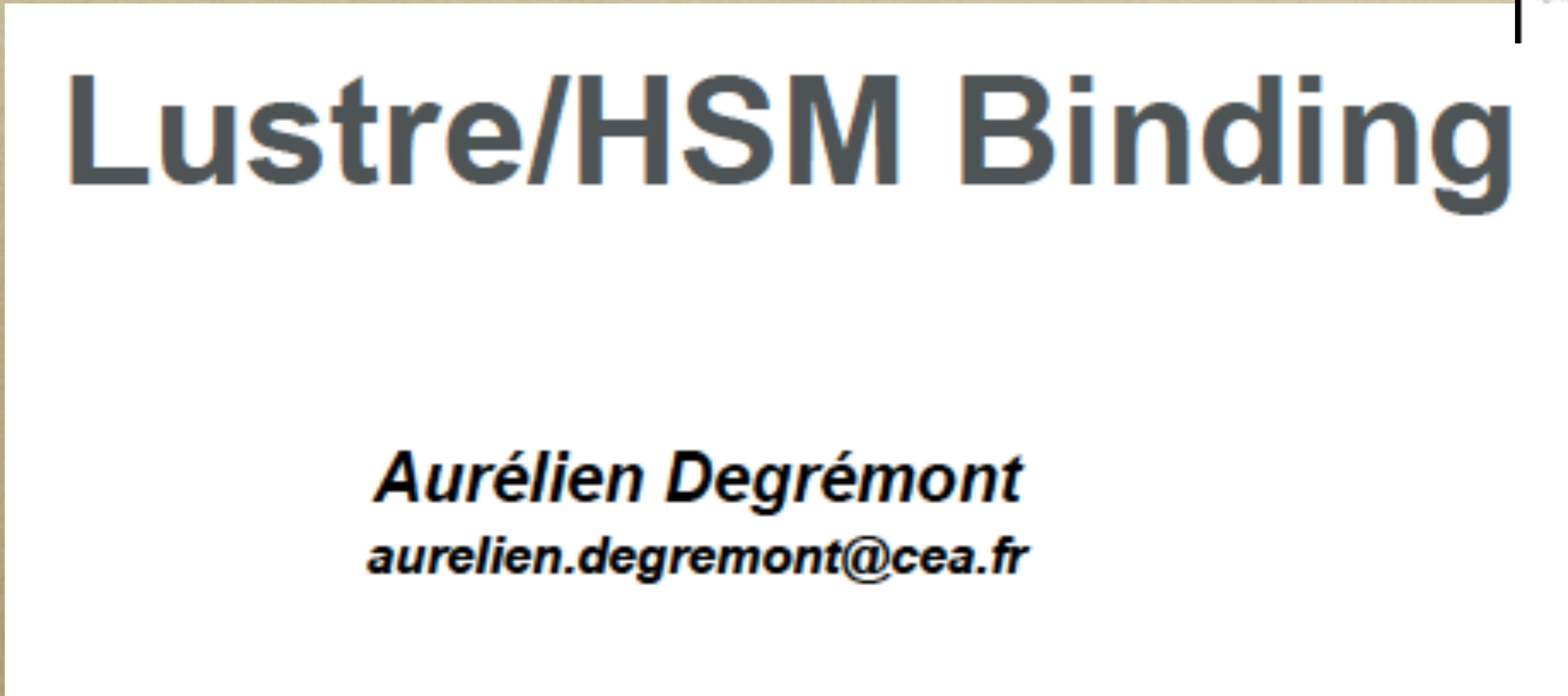
*8 Gbit/s

16*20 Gbit/s DDR

5*40 Gbit/s QDR IB

Lustre WAN @ 100GBit Testbed





Lustre/HSM Binding

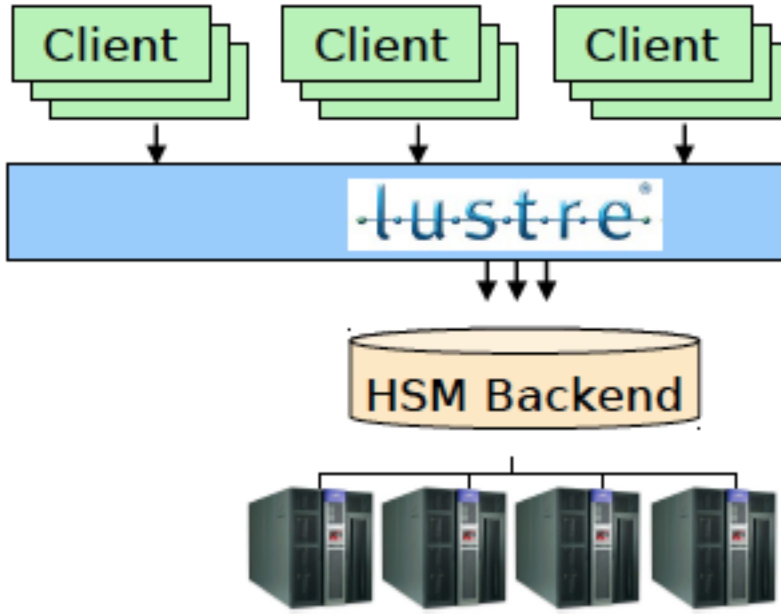
Aurélien Degrémont
aurelien.degremont@cea.fr

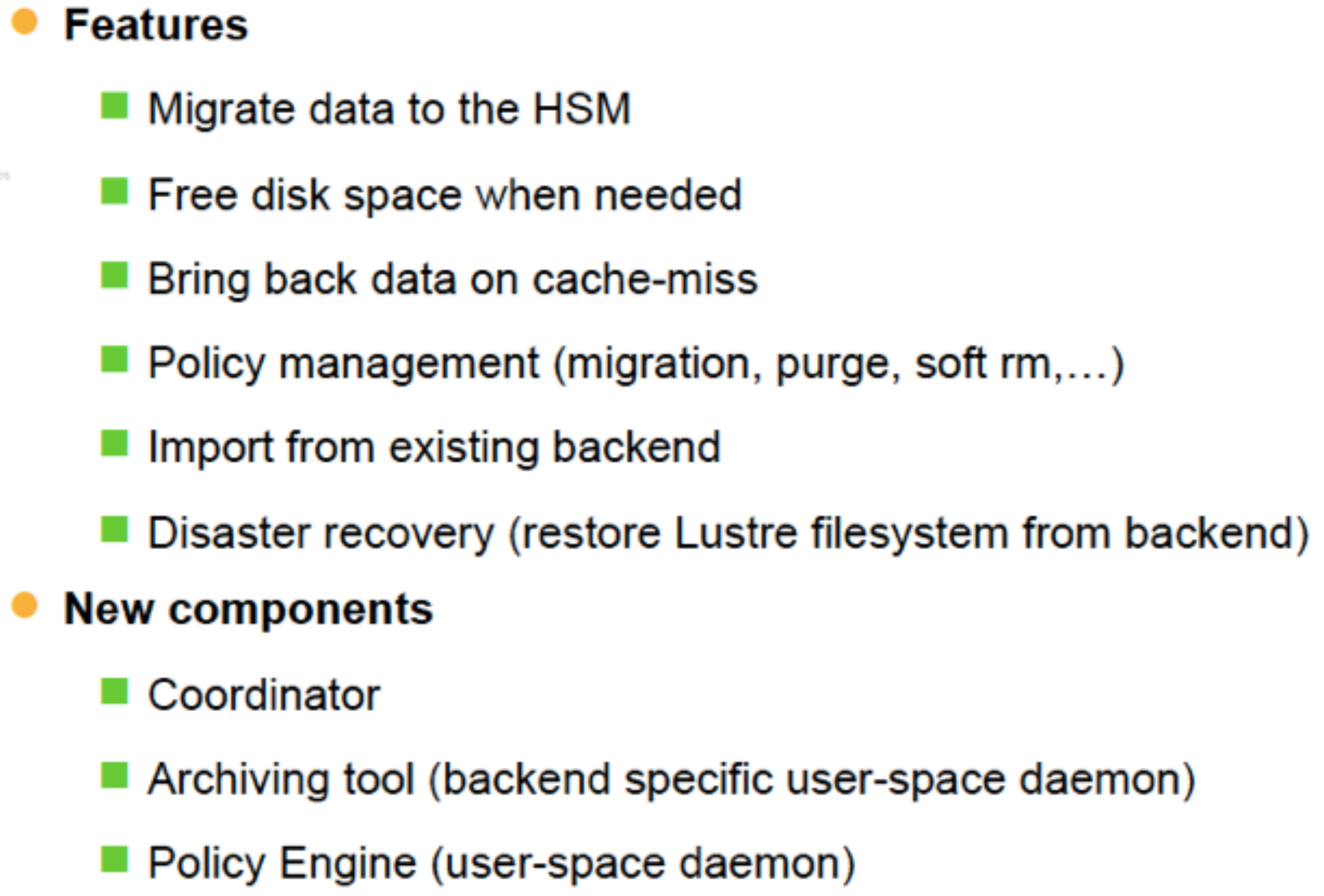
Lustre/HSM Binding

- *2007 – CFS times*
  - *Never ending Architecture*
- *2008-2009 – Sun era*
  - *Designing and Lustre internals learning*
- *2010 – Oracle times*
  - *Coding, hard landing*
- *2011 – Nowadays*
  - *Debugging, Testing, Improving*

Lustre/HSM Binding

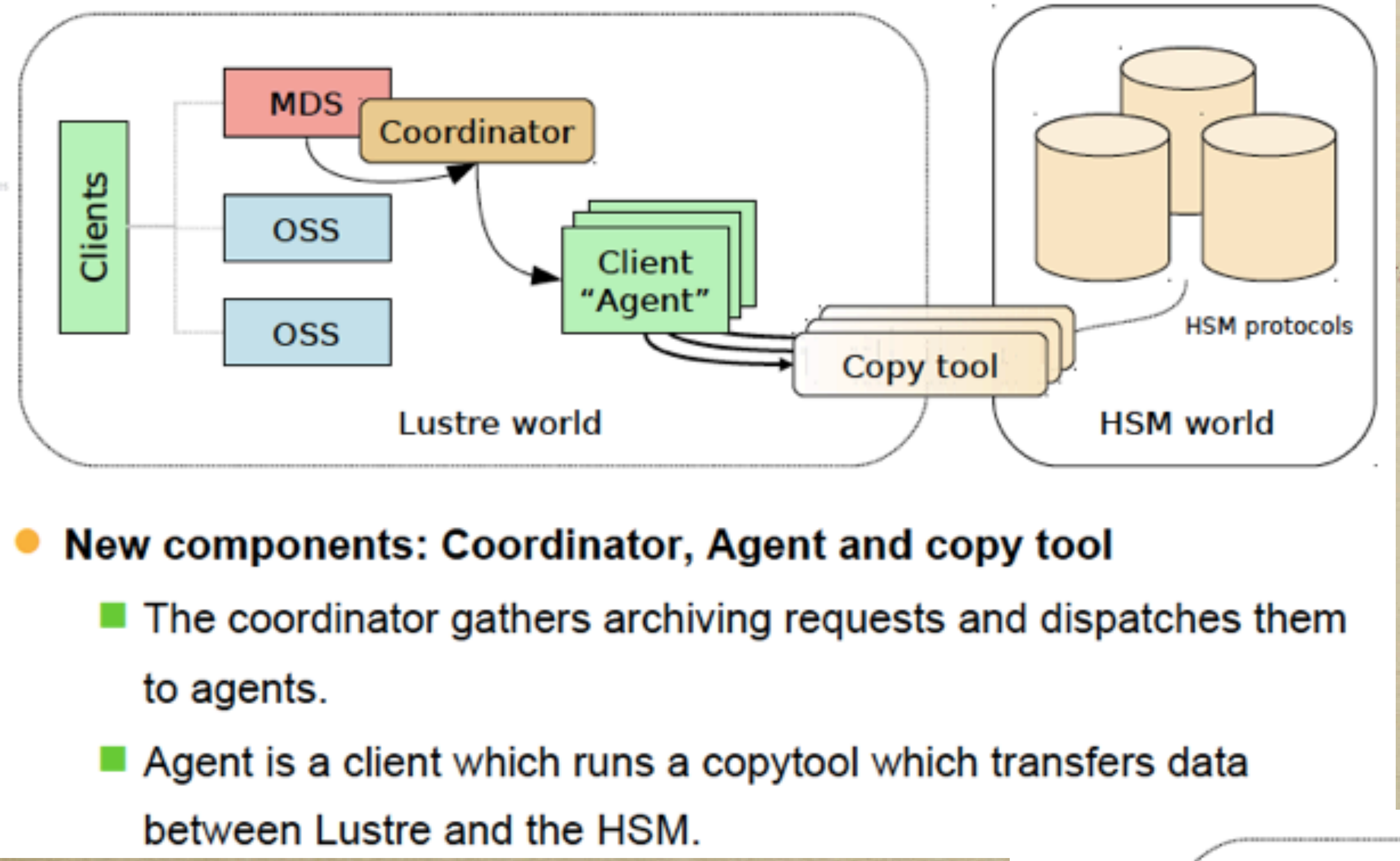
HSM seamless integration



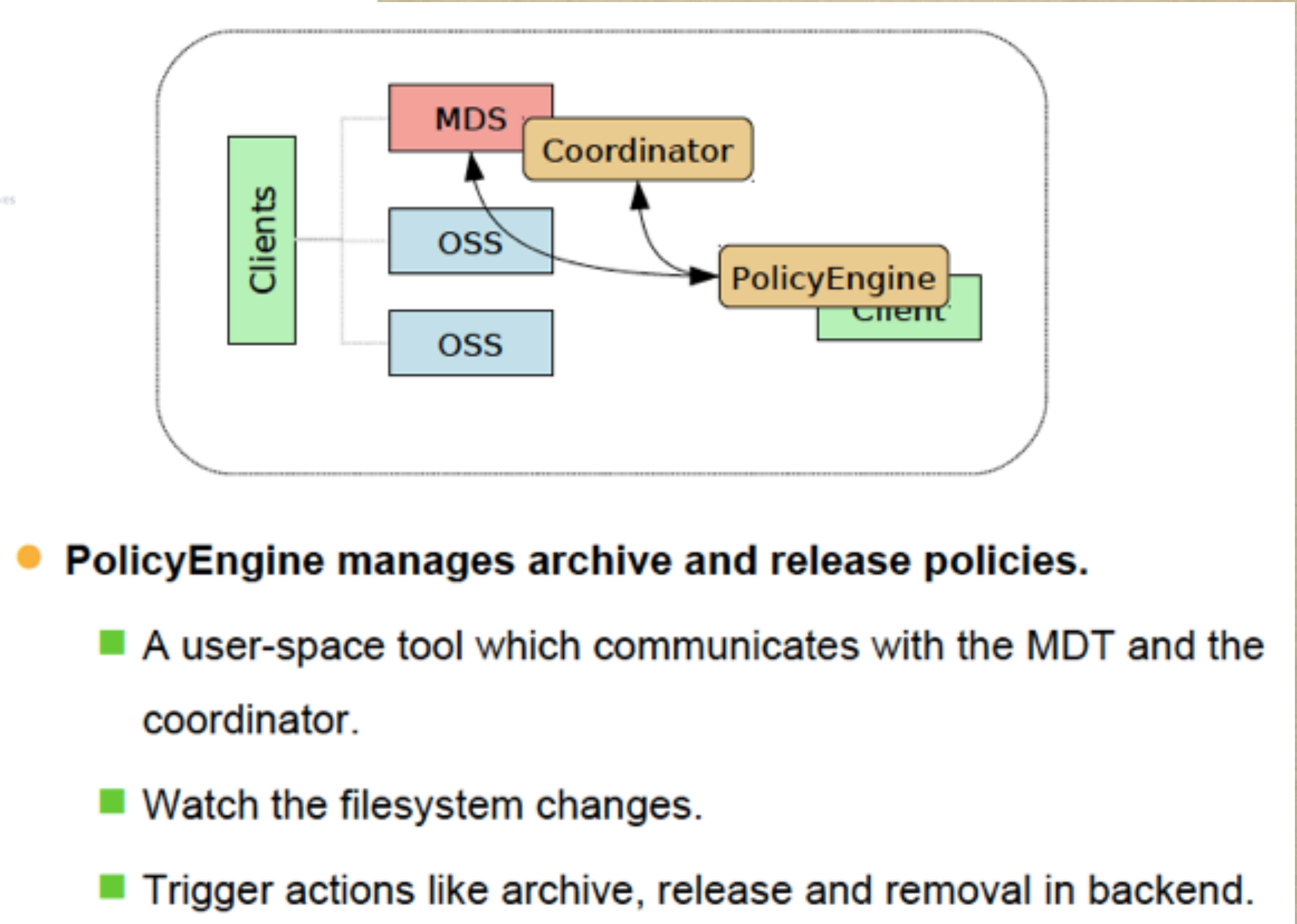


- **Features**
  - Migrate data to the HSM
  - Free disk space when needed
  - Bring back data on cache-miss
  - Policy management (migration, purge, soft rm,…)
  - Import from existing backend
  - Disaster recovery (restore Lustre filesystem from backend)
- **New components**
  - Coordinator
  - Archiving tool (backend specific user-space daemon)
  - Policy Engine (user-space daemon)

Lustre/HSM Binding



- **New components: Coordinator, Agent and copy tool**
  - The coordinator gathers archiving requests and dispatches them to agents.
  - Agent is a client which runs a copytool which transfers data between Lustre and the HSM.



- **PolicyEngine manages archive and release policies.**
  - A user-space tool which communicates with the MDT and the coordinator.
  - Watch the filesystem changes.
  - Trigger actions like archive, release and removal in backend.

Lustre/HSM Binding

- *Component: Copytool*
  - *It is the interface between Lustre and the HSM.*
  - *It reads and writes data between them. It is HSM specific.*
  - *It is running on a standard Lustre client (called Agent).*
  - *2 of them are already available:*
    - *HPSS copytool. (HPSS 7.3+). CEA development which will be freely available to all HPSS sites.*
    - *Posix copytool. Could be used with any system supporting a posix interface, like SAM/QFS.*
  - *More supported HSM to come*
    - *DMF*
    - *Enstore*

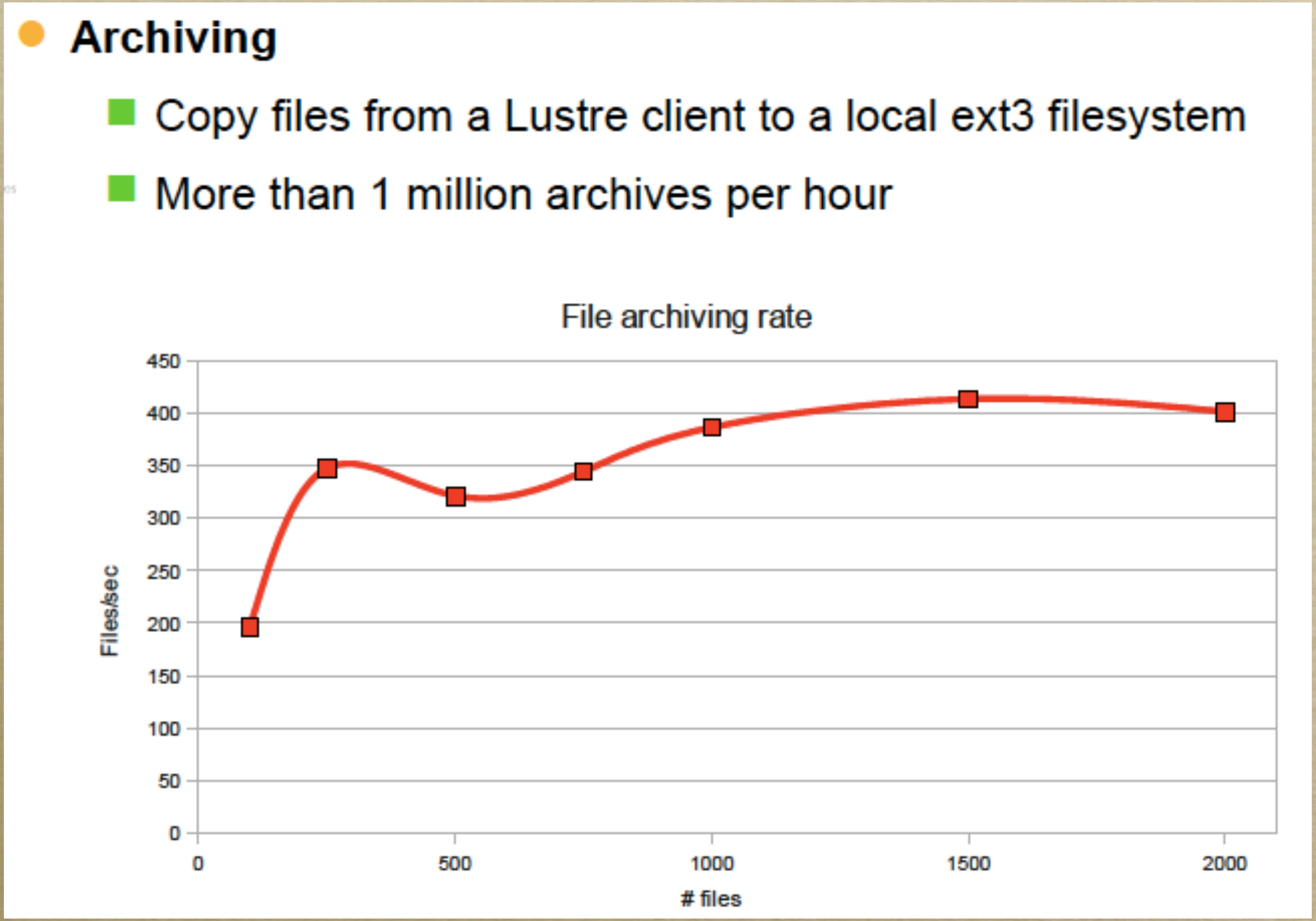
Lustre/HSM Binding

- *Component: PolicyEngine Robinhood*
  - *PolicyEngine is the specification*
  - *Robinhood is an implementation:*
    - *Is originately an user-space daemon for monitoring and purging large filesystems.*
    - *CEA opensource development: http://robinhood.sf.net*
  - *Policies:*
    - *File class definitions, associated to policies*
    - *Based on files attributes (path, size, owner, age, xattrs…)*
    - *Rules can be combined with boolean operators*
    - *LRU-based migr./purge policies*
    - *Entries can be white-listed*

Lustre/HSM Binding

- Component: Coordinator
  - MDS thread which "coordinates" HSM-related actions.
    - Centralizes HSM-related requests.
    - Ignore duplicate request.
    - Control migration flow.
    - Dispatch request to copytools.
    - Requests are saved and replayed if MDT crashes.
- View file states
  - lfs hsm_state <FILE>
  - $ lfs hsm_state /mnt/lustre/foo
  - /mnt/lustre/foo
    - states: (0x00000009) exists archived

Lustre/HSM Binding



- **Archiving**
  - Copy files from a Lustre client to a local ext3 filesystem
  - More than 1 million archives per hour

- *Will start code landing as soon as 2.2 branch is available*

ZFS/BTRFS & Lustre

- *Evaluate pooling at the filesystem layer*
  - *Avoids expensive RAID controllers*
  - *Provides additional features*
  - *Copy-on-write*
  - *Built-in data integrity*
  - *Very large filesystem limits*
  - *Late 2011 requirement:*
    - *50PB, 512GB/s – 1 TB/s*
    - *At a price we can afford*
  - *COW sequentializes random writes*
    - *No longer bound by drive IOPS*
  - *Zero fsck time. On-line data integrity and error handling*

Australian NCI National Facility

- *Current machine “vayu”*
  - *~1500 nodes, ~12k Nehalem cores*
  - *26 OSS's, 4 MDS's*
- *Root on Lustre – Why?*
  - *Simplicity*
    - *Fewer things to fail*
    - *No NFS or local disks involved Reliability and Scalability*
  - *Use centralised scalable and reliable hardware*
    - *If Lustre is down then jobs are hung anyway. May as well put the OS there too*
  - *Maintainability*
    - *One rsync from the master OS image to the OS image on Lustre updates every node immediately*
    - *Unlimited space for OS packages, OS variations, ...*

Australian NCI National Facility

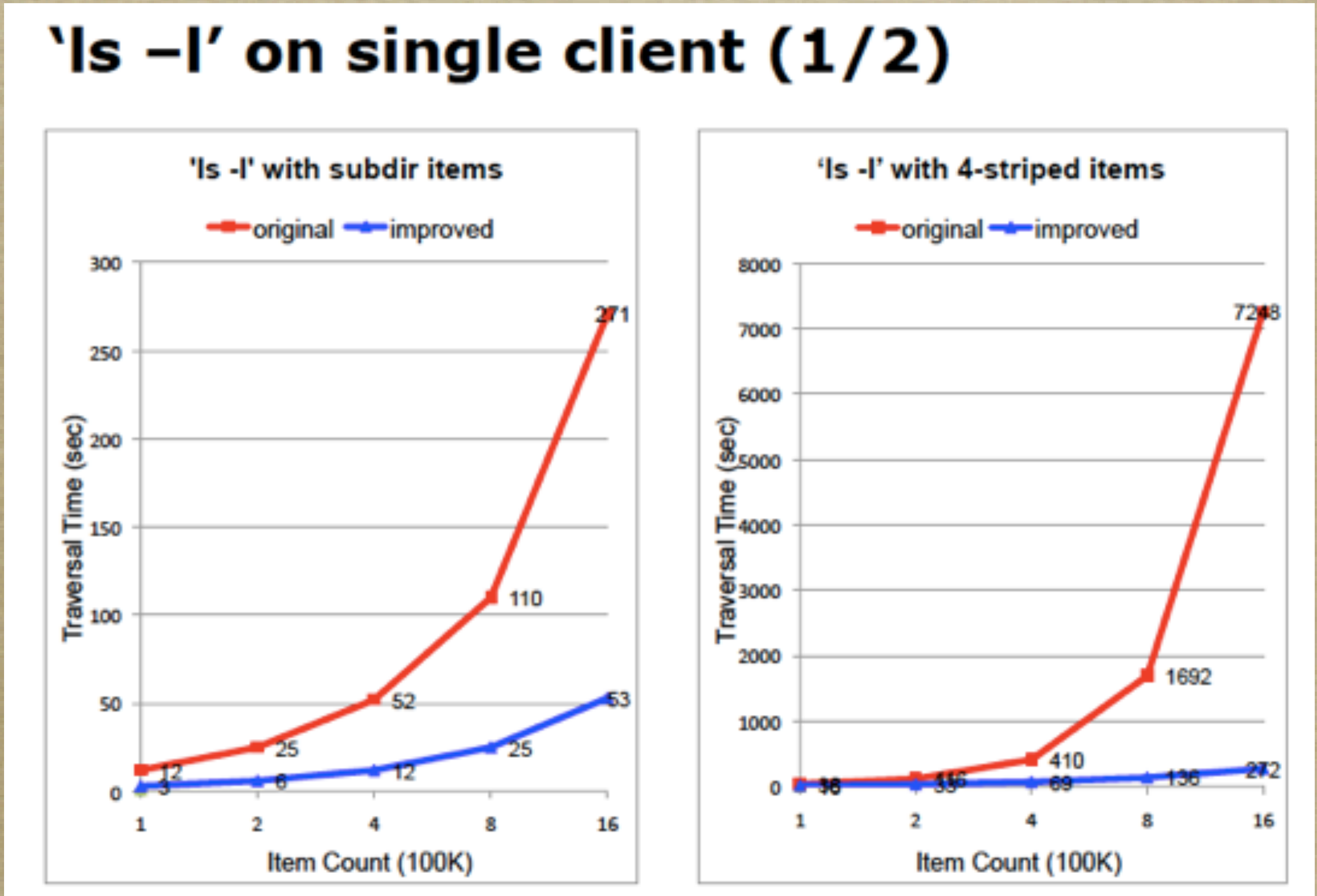
- *Metadata Speed – The Problem:*
  - *Very slow "ls -l"*
  - *Uncached "ls -altrR ~" runs at ~100 files/second*
  - *Client-side caches help, but only when nodes aren't busy*
  - *Daily rsync backups taking >24hrs*
- *Metadata Speed – Root Cause:*
  - *MDS? No*
  - *Loads low*
  - *All fs data fits entirely in ram*
    - *MDT's are a 4k i/o write-only media after a while*
- *OSS's? Yes*
  - *Very busy OSS's*
  - *Streams to read and write-through caches aggressively pushing ldiskfs inodes/dentries out of OSS ram*

Australian NCI National Facility

- *Metadata - vfs_cache_pressure*
  - *What is vfs_cache_pressure?*
    - *Balance between pages (data) cached and inodes/dentries (ldiskfs metadata) cached*
    - *=100 by default =0 means NEVER reclaim any inodes/dentries*
      - *Dangerous! Scary! Can OOM!*
  - *But…*
    - *inodes are 912 bytes, dentries are 216 bytes - Tiny! 1G of slab ram on 1 OSS ~= 1M files*
    - *Low mem OSS's shouldn't use read or write-through caches*
    - *inode/dentry usage grows slowly with fs*
- *Result*
  - *20 to 40x speedup of "ls -lR" and >10x speedup of rsync backups*
  - *Typical "ls -altrR ~" on an un-cached client is ~4k files/s (when client cached is 32k files/s)*
  - *Repeatable day to day. ie. Caches are being preserved*
  - *Problem solved!*

Few other report

- *Work on going to increase the number of OSTs stripes on each file (at the moment it is 160 OSTs)*
  - *This is related also to the max size of a single file: 320TB*
  - *and to the max bandwidth on a single file*
- *Work on going to increase the performance in creating file:*
  - *at the moment it is about 20’000/s*
- *Metadata improvement by means of async&bulk RPC call*



‘ls –l’ on single client (1/2)

Conclusions & personal ideas

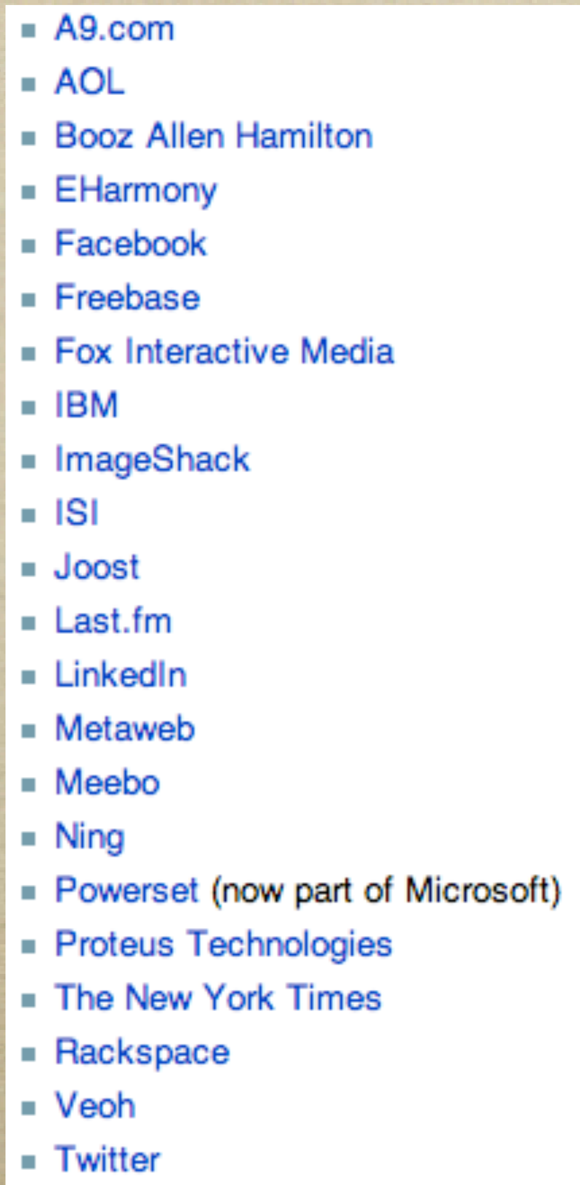
- *Open Source Community is growing around medium-big company*
- *Big Computational centres are directly involved into deploying and developing*
  - *Often this centres have different requirement than HEP community*
- *2.1 it is almost done*
  - *it will available around summer => this will be needed in case of SL6 migration*
- *the real milestone will be 2.2 release*
  - *if this will be released, public available and supported this will be a “long term release”*
    - *expected at the end of this year*

Emergency plan??

- *Fall back to few ad-hoc (HEP) solution:*
  - *dCache? EOS/Xrootd?*
    - *Cons:*
      - *Support issues*
      - *Users communities*
      - *Often users requires (good) posix compliance*
- *If possible, it will be preferable to use widely used, open source, well maintained solution ...*

Emergency plan??

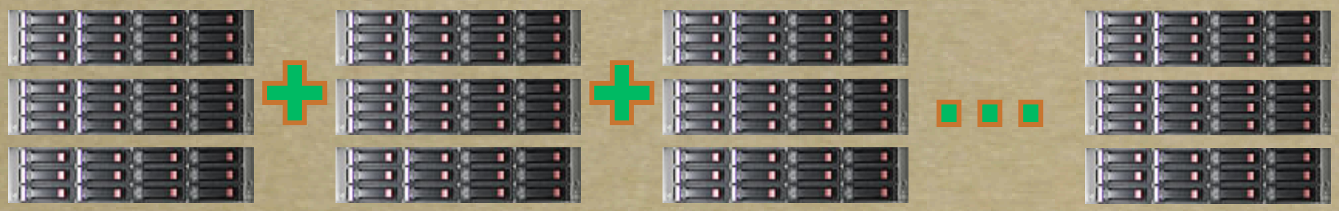
- ***Hadoop (HDFS):***
  - *It is developed till 2003 (born @google)*
  - *It is a framework that provide: file-system, scheduler capabilities, distributed database*
  - *Fault tolerant*
    - *Data replication*
    - *DataNode failure is ~transparent*
    - *Rack awareness*
  - *Highly scalable*
  - *Using FUSE => few posix call supported*
    - *roughly "all read operation" and only "serial write operations"*
  - *Web interface to monitor the HDFS system*
  - *Java APIs to build code that is "data location aware"*
  - *CKSUM at file-block level*
  - *HDFS RAID (2.2 space used == 3 copies)*



- A9.com
- AOL
- Booz Allen Hamilton
- EHarmony
- Facebook
- Freebase
- Fox Interactive Media
- IBM
- ImageShack
- ISI
- Joost
- Last.fm
- LinkedIn
- Metaweb
- Meebo
- Ning
- Powerset (now part of Microsoft)
- Proteus Technologies
- The New York Times
- Rackspace
- Veoh
- Twitter

Emergency plan??

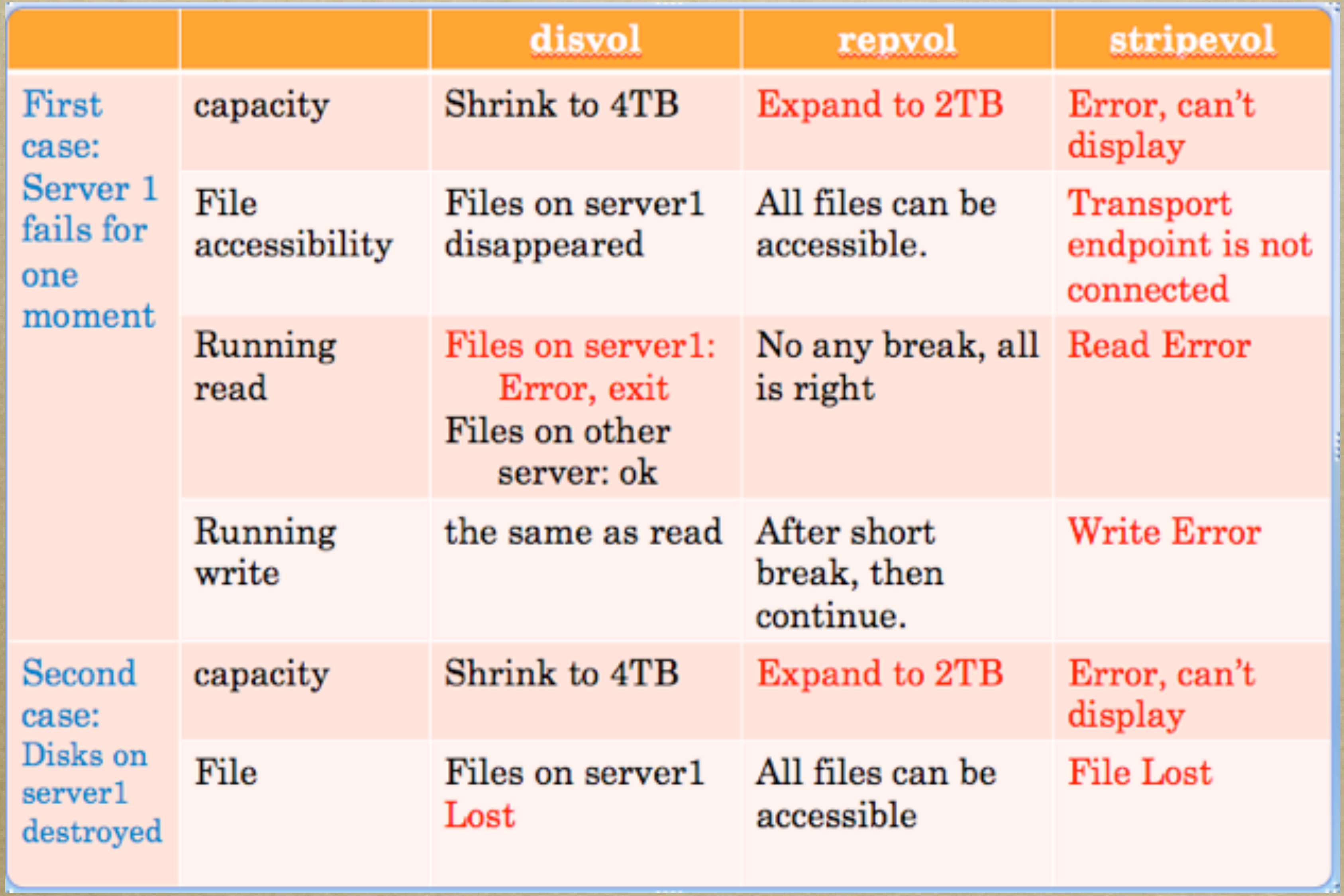
- ***GlusterFS:***
  - *It is a scalable open source clustered file system*
  - *It aggregates multiple storage bricks over Ethernet or Infiniband RDMA interconnect into one large parallel network file system*
  - *offers a global namespace*
  - *Focus on scalability, elasticity, reliability, performance, ease of use and manage, …*
  - *More scalable, reliable*
  - *No Metadata server with elastic HASH Algorithm*
  - *More flexible volume management (stackable features)*
  - *Elastic to add, replace or remove storage bricks*
  - *Automatic file replication, Snapshot, and Undelete*
  - *Fuse-based client*
    - *Fully POSIX compliant*
    - *NO ACL (StoRM problem)*



N × Performance & capacity

Emergency plan??

- ***GlusterFS crash test:***
  - *Two cases*
  - *Storage server or network fails for one moment, then recovers*
  - *Disk is destroyed and all data in the disk is lost permanently*
- *Different types of volume (distributed, striped, replicated volumes) and running operations (read, write) have different affects in the two cases*
- *Running operations mean that one is reading or writing files when storage server or network fails*



		disvol	repvol	stripevol
First case: Server 1 fails for one moment	capacity	Shrink to 4TB	Expand to 2TB	Error, can't display
	File accessibility	Files on server1 disappeared	All files can be accessible.	Transport endpoint is not connected
	Running read	Files on server1: Error, exit Files on other server: ok	No any break, all is right	Read Error
	Running write	the same as read	After short break, then continue.	Write Error
Second case: Disks on server1 destroyed	capacity	Shrink to 4TB	Expand to 2TB	Error, can't display
	File	Files on server1 Lost	All files can be accessible	File Lost

Emergency plan??

- ***GlusterFS vs HDFS comparisons:***
  - ***!!!my personal view!!! (from 0 to 10)***

	GlusterFS	*Hadoopfs*
Resilient to failure	*6*	*8*
Posix compliance	*8*	*4*
Performance	*9*	*6*
Community	*6*	*8*
Scalability	*6*	*8*
Metadata performance	*6*	*8*